RCA PF030016AD

CITED BY APPLICANT

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 9/46, H04L 29/06

A1

(11) International Publication Number: WO 99/34286

(43) International Publication Date: 8 July 1999 (08.07.99)

(21) International Application Number: PCT/US98/27420

(22) International Filing Date: 23 December 1998 (23.12.98)

(30) Priority Data:
60/068,893 29 December 1997 (29.12.97) US

(71) Applicant: POSTX CORPORATION [US/US]; Suite 2, 10455 Bandley Drive, Cupertino, CA 95014 (US).

(72) Inventors: VENKATRAMAN, R., C.; 1031 Harlan Drive, San Jose, CA 95129 (US). CHERN, Vincent, Min-Hao; 7th floor, 4225 Executive Square, La Jolla, CA 92037 (US). NANJA, Sekaran; 5824 Chambertin Drive, San Jose, CA 95118 (US).

(74) Agents: JAKOPIN, David, A. et al.; Pillsbury Madison & Sutro, LLP, 1100 New York Avenue N.W., Washington, DC 20005 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

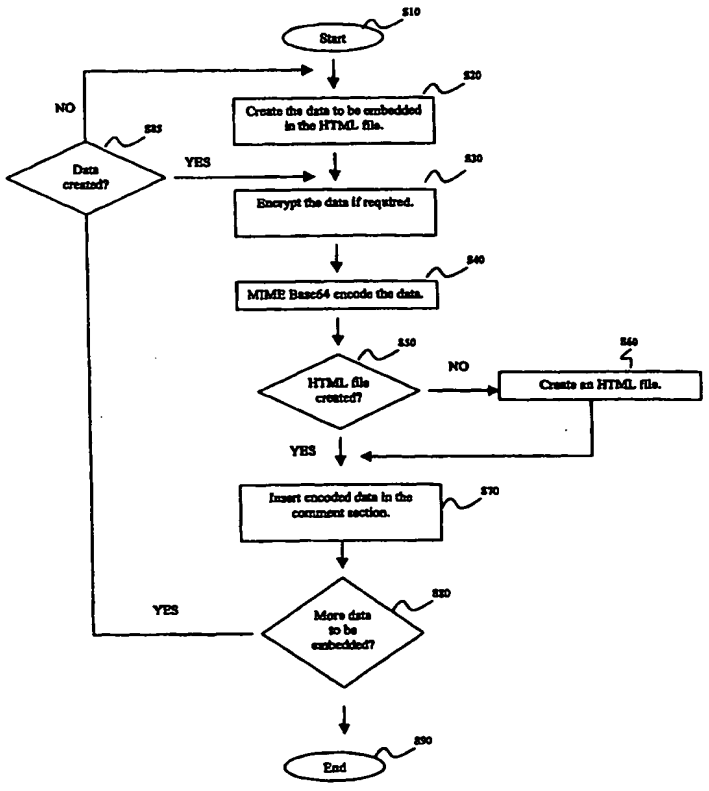
With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD AND APPARATUS CAPABLE OF EMBEDDING, EXTRACTING AND PROCESSING DATA WITHIN A FILE HAVING AN HTML FORMAT

(57) Abstract

A method and apparatus is provided that is capable of distributing data within a file having an HTML format. A first process creates the data, encodes it in a known format such as MIME Base64, and embeds the data within a comment section of an HTML file. A second process, preferably implemented as a browser plug-in application, is used to extract the data from the HTML file and store the data in a separate data file. A third process creates an HTML file that launches an applet that uses the data, and contains applet parameters including the paths to the separate data files in which the extracted data resides.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

METHOD AND APPARATUS CAPABLE OF EMBEDDING, EXTRACTING AND PROCESSING DATA WITHIN A FILE HAVING AN HTML FORMAT

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to digital data communications, and more particularly, to a method and apparatus for distributing data and programs between networked computers via files having an HTML format in a secure and self-executing fashion.

2. Description of the Related Art

Recently, Internet web browsing has become nearly ubiquitous and web browser applications have become standard equipment on desktop computing platforms.

Most web browsers are capable of loading files having the well-known HTML format, described generally in Dave Raggett, "HTML 4.0 Reference Specification," Dec. 1997, available at http://www.w3.org/TR/REC-html40. Most information shared on the Internet is supplied via data files having the HTML format.

Most web browsers also support plug-in applications. Such plug-in applications can include programs that are written in the Java programming language. For the Netscape 4.0 browser environment (trademark of Netscape Communications Corp. of Mountain View, Calif.), plug-ins are described generally in "Plug-Ins Documentation," Nov. 1997, available at http://developer.netscape.com/library/documentation/index.htm. As a result, browsers have become platforms on which applications written in the Java programming language can be executed. Since browsers typically support a number of operating environments, this provides a means by which an application written in one common programming language can be executed with the same results across a variety of operating environments.

The present invention aims at leveraging the capabilities of interchanging and processing data via HTML files and conventional web browsers in new and useful ways.

SUMMARY OF THE INVENTION

An object of the present invention is to distribute data and programs between networked computers in an efficient and fast manner.

It is another object of the invention to distribute data and means for presenting the data in an integrated fashion.

It is another object of the invention to distribute data and means for presenting the data in a uniform way across a variety of operating environments.

A method and apparatus is provided that fulfills these and other objects and is capable of distributing data within a file having an HTML format. A first process creates the data, encodes it in a known format such as MIME Base 64, and embeds the data within a comment section of an HTML file. A second process, preferably implemented as a browser plug-in application, is used to extract the data from the HTML file and store the data in a separate data file. A third process creates an HTML file that launches an applet that uses the data, and contains applet parameters including the paths to the separate data files in which the extracted data resides.

BRIEF DESCRIPTION OF THE DRAWINGS

The details of the invention, as well as the best mode for practicing it, will become apparent to those skilled in the art by reference to the following detailed description, taken in conjunction with the appended drawing figures, in which:

Figure 1 is a block diagram illustrating a configuration in accordance with the principles of the invention;

Figure 2 is a flowchart illustrating a process of creating encoded data and embedding it in an HTML file in accordance with the principles of the invention;

Figure 3 illustrates the format of an HTML file containing embedded data such as that created in the process illustrated in Figure 2;

Figure 4 is a flowchart illustrating a process of extracting embedded data from an HTML file such as that created in the process illustrated in Figure 2 in accordance with the principles of the invention;

Figure 5 is a flowchart illustrating a process of launching an applet that uses embedded data such as that extracted in the process illustrated in Figure 4 in accordance with the principles of the invention;

Figure 6 illustrates the format of an HTML file containing instructions for launching an applet such as that created in the process illustrated in Figure 5;

Figure 7 illustrates a specific example of the format of an HTML data file created in accordance with the invention; and

Figure 8 illustrates a specific example of the format of an HTML file containing instructions for launching an applet created in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

As shown in Figure 1, sending side host 2 includes a computer 10-A, and a user interface 20-A. Sending side computer 10-A includes a CPU 12-A, an executable RAM 14-A, and a mass storage drive 16-A. Receiving side host 4 includes a computer 10-B, and a user interface 20-B. Receiving side computer 10-B includes a CPU 12-B, an executable RAM 14-B, and a mass storage drive 16-B. Computers 10-A and 10-B are functional as is conventionally known to execute programs loaded into RAM 14-A and 14-B, which programs contain instructions that are performed by CPU 12-A and 12-B. Such programs can be loaded into RAM 14-A and 14-B from mass storage drive 16-A and 16-B. Such programs can also create, manipulate and store data into mass storage drive 16-A and 16-B, as well as cause data to be presented in a window 18-A and 18-B on user interface 20-A and 20-B and receive user selections from user interface 20-A and 20-B. Preferably, at least the receiving side computer is loaded with a Java-enabled browser.

It should be understood that, although shown separately here for clarity, a host can act as a sending side host in one transaction and as a receiving side host in another transaction within the principles of the invention.

In accordance with the principles of the invention, sending side host 2 and receiving side host 4 are capable of sending and receiving files via a network 6, preferably as attachments to e-mail messages. However, many other means of sharing files can be employed. Such data files can include HTML data file 30, created in accordance with the principles of the invention as described in more detail below. Network 6 can be any data communication network, public or private. Receiving side host 4 is preferably loaded with browser programs that support Java-based applets and plug-ins written in Java and/or native programming languages.

In further accord with the principles of the invention, receiving side host 4 is capable of sending and receiving files from a server 8. Preferably, files on server 8 are accessible on the World Wide Web via conventional browsers and download tools.

An example of a first process used to create and embed data in an HTML data file 30 in accordance with the principles of the invention will now be described with reference to the flowchart depicted in Figure 2.

Preferably, the first process is implemented as a program loaded into executable RAM 14-A on sending side computer 10-A, which program contains instructions that are executable by CPU 12-A in accordance with the processing steps described below with reference to Figure 2.

It should be understood that all of the process steps can be implemented together as a single standalone program or module that executes using certain input parameters, or they can be separately implemented using a combination of readily-available commercial tools and/or specially written tools or modules.

As shown, upon starting (step S10), processing begins by creating the data that is to be sent to the receiving side host 4 (step S20). The data to be embedded can be, for example, an executable file, such as a Java applet, an image file such as a GIF file, an Envelope Data File such as that described in co-pending U.S. Application No. 08/845,722 and explained in more detail below, or a native code module. Although shown as a step within the current processing, it should be understood that the data can actually be a previously created data file or object, in which case this step simply includes identifying and locating the data to be embedded.

The created data can then be encrypted, if required by the user (step S30). Such encryption can be performed using encryption technology such as RC4, a trademark of RSA Data Security, Inc. of Redwood City, CA. In step S40, the data is encoded using the MIME Base64 format. Whether an HTML file has already been created is determined in step S50. If not, the HTML file is created (step S60). Preferably included in the step of creating the HTML file are steps of adding an instruction within the file that will cause a plug-in application to begin executing so as to extract the embedded data. Such a plug-in application will be described in more detail below. Also preferably included in this step is an HTML script that determines whether a correct version of the plug-in application exists locally within the receiving computer, and if not, downloads the correct version of the plug-in application.

Finally, the encoded data is embedded in the comment section of the HTML file (step S70). This is done by creating a comment section and inserting the encoded data therein. This step may also include creating a header for the encoded data in the comment section, which header may include information about the encoded data and whether the encoded data is encrypted. It should be further noted that care should be taken to insure that the total size of the HTML file including the encoded data is within the limits of conventional browser applications.

If more data is left to be embedded (determined in step S80), for example, if the data to be embedded is a list of separate data objects, processing returns to step S85 where it is determined whether the remaining data is already created. If not, processing returns to step S20, otherwise processing returns to step S30 and loops until all the data has been embedded.

Figure 3 illustrates the format of a HTML file created in accordance with the first process described above. As shown, it includes a script section 50, a plug-in execution section 52, and comment sections 54-1...54-N.

Represented as pseudocode, script section 50 preferably includes HTML instructions to determine whether a correct version of the plug-in application for extracting the embedded data exists locally to the recipient computer. If not, the correct plug-in application is downloaded, for example, from a path specified in the script instructions.

Plug-in execution section includes HTML instructions for causing the plug-in application to be executed.

Comment sections 54-1...54-N are separate comment sections for each of the data objects embedded in accordance with the processing steps outlined above.

An example of a second process used to extract data embedded, via a process such as that described above, in an HTML file in accordance with the principles of the invention will now be described with reference to the flowchart depicted in Figure 4.

Preferably, the second process is implemented as a program loaded into executable RAM 14-B on receiving side computer 10-B, which program contains instructions that are executable by CPU 12-B in accordance with the processing steps described below. Such a program is preferably implemented as a plug-in application to a browser. The plug-in application is executable code, for example, Java byte code and/or native code. When the browser encounters an HTML instruction requesting its execution, the plug-in application is loaded into the executable memory of the computer and executed. The browser includes a plug-ins directory which the browser searches when the application is requested.

As shown, after starting (step S100), processing begins by reading the HTML page that initiated the program and saving it to a file (step S110). Next, the HTML file is read and the next embedded data within a comment is extracted (step S120). The embedded data is then decoded out from the MIME Base64 format (step S130). If the data was encrypted, it is decrypted to its original state in step S140. Whether the data needs to be decrypted may be determined from a header within the embedded data, or it may indicated by a flag or variable within or external to the HTML file.

Finally, the data is moved to its proper directory (step S150), which directory may be predetermined and pre-existing or may be dynamically assigned and created at run-time. A determination is next made whether the end of the HTML file has been reached (step S160),

for example, whether no more comment sections containing embedded data remain. If so, the processing ends (step S170); otherwise processing loops back to step S120 and the above-described processing is repeatedly performed until the end of the HTML file is reached.

An example of a third process used to launch an applet that uses data embedded and extracted, via processes such as that described above, in accordance with the principles of the invention, will now be described with reference to the flowchart depicted in Figure 5.

Preferably, the third process is implemented as a program loaded into executable RAM 14-B on receiving side computer 10-B, which program contains instructions written in the Java programming language that are executable by CPU 12-B in accordance with the processing steps described below. Such a program is preferably implemented contiguously with the program containing the second process described above, and may be part of the same executable software as the plug-in application that contains the second process.

As shown, after starting (step S200), processing begins by creating an HTML page that contains an instruction to launch the applet (step S210). The input parameters for the applet are the file paths of the embedded data that were extracted from the HTML file using the second process described above, which file paths are preferably communicated to the third process by the second process. Next, the third process asks the browser to load the HTML page created in step S210 (step S220) by, for example, a call to a plug-in Application Programming Interface (API) of the browser or a Java API, and processing ends (step S230). As should be apparent, when the HTML page created in step S210 is loaded by the browser, the applet is caused to execute.

The executable software containing the applet may be downloaded at the same time as the program containing the second and third process, or it may already be located in the mass storage 16-B of the receiving side computer 10-B, and loaded into memory upon a launching instruction. Alternatively, the applet may be contained in one of the extracted data files included in the HTML file that was stored in a directory on the receiving side computer 10-B during the second process.

Figure 6 illustrates the format of a HTML file created in accordance with the third process described above. As shown, it includes an applet description section 60 and applet parameter sections 62-1...62-N. Of note within the applet description section 60 is the path to the archive of Java classes xxx.jar, as well as the identifier of the applet yyy.class within the container of classes.

In operation, with reference to Figure 1 and the preceding descriptions, when sending side host 2 desires to send data to receiving side host 4, the data is encoded and embedded in an HTML data file 30 in a program executing instructions on the sending side host computer 10-A in accordance with the first process described above. The sending side host 2 then sends an e-mail message to receiving side host 4, with HTML data file 30 as an attachment. On receiving side host 4, the e-mail message is received and read by an e-mail program executing on receiving side host computer 10-B, and a browser is launched to view HTML data file 30 attached to the e-mail. On a computer running a Windows 95 (trademark of Microsoft, Corp. of Redmond, Wash.) environment, for example, double-clicking on a screen object representing the e-mail attachment will automatically launch a browser application in accordance with the HTML file extension. The browser causes the second process, launched as a plug-in application under the browser and executing on receiving side host computer 10-B, to extract the data and store it on mass storage drive 16-B as described above. The third process is then invoked to launch an applet that uses the data, which applet can include presenting the data on user interface 20-B.

An example of the invention will now be described with reference to Figures 7 and 8.

In addition to web browsers, e-mail has also become nearly ubiquitous in recent times. Notorious problems with e-mail still remain, however, and include lack of homogeneity in presentation of information, lack of security and lack of verification of receipt. An e-mail enhancement that solves these problems, among others, is described in the co-pending application of Venkatraman et al., U.S. Patent Application No. 08/845,722, filed April 25, 1997, the contents of which are incorporated herein by reference.

In the e-mail enhancement described in the co-pending application, a container file, or Envelope Data File, is created that contains one or more component objects or "Vixels." The component objects may include data and one or more defined "User Actions" which, for example, cause the presentation of the data in a desired manner. The Envelope Data File is included as an e-mail attachment file and sent to an intended recipient, where the envelope contents can be selectively presented and manipulated.

The co-pending application also describes creation executable software for creating the Envelope Data File, on a sending side computer for example, and recipient executable software for presenting the data in the desired manner, on a receiving side computer for example. The

recipient executable software can be included in the e-mail attachment sent to the recipient, thereby transporting the capability of presenting the data to the recipient in unique ways.

In conjunction with the present invention, the creation executable software, for example executing on a sending side computer such as 10-A, includes a first process such as that described in the invention. Accordingly, the creation executable software, while creating the Envelope Data File, encodes it and embeds it into an HTML data file such as HTML data file 30. The recipient executable software, comprised of a number of recipient executable files determined by the different number of images, sounds, and functional attributes inserted into the envelope container, for example, can also be encoded and embedded within the HTML data file by a first process such as that described in the present invention.

As shown in Figure 7, an example of an HTML data file created by a first process such as that described above and in conjunction with the co-pending application includes an HTML script section 70, a plug-in execution section 80, a first embedded data section 90, and a second embedded data section 92.

HTML script section 70 is an implementation specific for a Netscape (trademark of Netscape, Inc. of Mountain View, Ca.) browser environment. It includes a download section 72, in which a correct version of the extraction process plug-in is downloaded from an Internet site designated by field 76. Field 74 identifies the path to the extraction process plug-in. It should be noted that the principles of the invention are applicable to other browser environments and standards such as Internet Explorer (a trademark of Microsoft Corp. of Redmond, Wash.), and those skilled in the art will recognize the changes necessary to adapt to such other environments and standards.

Plug-in execution section 80 includes an HTML instruction to run the plug-in application, as identified in field 82. Of particular note is that the path of the plug-in application indicated in field 82 is the same as that indicated in field 74. At this point of processing the HTML file by the browser, the correct version of the plug-in application should exist in the location indicated in field 82.

First embedded data section 90 includes a header 94 and a footer 96 acting as delimiters and identifiers of the encoded data within first embedded data section 90. In this example, the first embedded data section 90 includes an Envelope Data File as described in the co-pending application.

Second embedded data section 92 includes a header 98 and a footer 100 acting as delimiters and identifiers of the encoded data within second embedded data section 92. In this example, the second data section 92 includes data relating to a "Vixel" as described in the co-pending application.

An HTML file for launching an applet in accordance with the illustrated example of the invention is shown in Figure 8.

As described above, in this example of the invention, operating in conjunction with the e-mail enhancement described in the co-pending application, the first process executes as part of the creation executable software as described in the co-pending application. The second and third processes, for example executing on the receiving side computer 10-B, provide a means for managing the transfer of envelope container data and executables for uniquely presenting data in accordance with the principles described in the co-pending application. The applet, as described in the present invention, corresponds to the recipient executable software as described in the co-pending application, which applet can include instructions for presenting different images, sounds and functional attributes in accordance with the different types of objects included in the envelope data.

Accordingly, the HTML file illustrated in Figure 8, created by a third process as described in the present invention, includes an applet description section 64 and applet parameter sections 66-1...66-N. Of note within the applet description section 64 is the path to the archive of Java executables EvelopeApplet.jar, as well as the identifier of the applet EnvelopeApplet.class within the container of executables. Of further note within the applet parameter sections 66-1...66-N is the path to the Envelope Data File included in the "envelopeName" parameter.

Although the present invention has been described hereinabove with reference to the preferred embodiments thereof, those skilled in the art will appreciate that various substitutions and modifications can be made to the examples provided. For example, although the first, second and third processes have been described above as implemented by instructions executing on a processor, it should be apparent that various combinations of hardware and software components can be used to implement these processes.

Accordingly, such substitutions and modifications can be made while remaining within the proper scope and spirit of the invention as defined in the appended claims.

We Claim:

1. A method for distributing data between a sending side host and a receiving side host, comprising:

preparing data on said sending side host;

embedding said data within a comment section of an HTML file;

sending said HTML file to said receiving side host;

extracting said embedded data from said comment section of said HTML file;

preparing a local copy of said extracted data on said receiving side host.

2. A method as defined in claim 1, further comprising:

launching an applet on said receiving side host that uses certain portions of said extracted data.

3. A method as defined in claim 2, wherein said step of preparing said local copy includes locating said extracted data at a file path on said receiving side host, said step of launching said applet including passing a parameter including said file path of said local copy to said applet.

4. A method as defined in claim 3, wherein said step of launching said applet includes:

creating a second HTML file with an instruction to launch said applet;

inserting said parameter in said second HTML file; and

invoking an application to load said HTML file.

5. A method as defined in claim 1, wherein said step of preparing said data includes:

encrypting said data if required by a user; and

encoding said data in a predetermined format.

6. A method as defined in claim 5, wherein said step of preparing said local copy of said extracted data includes:

decoding said data in accordance with said predetermined format; and

decrypting said data if said decoded data is encrypted.

7. A method as defined in claim 1, further comprising:

determining whether a process for performing said extracting step exists on said receiving side host; and

downloading said process from an external source if said process does not exist on said receiving side host.

8. A method as defined in claim 2, wherein said applet is contained within certain other portions of said extracted data.

9. An HTML file created by the method of claim 1.

10. An apparatus for distributing data between a sending side host and a receiving side host, comprising:

means for preparing data on said sending side host;

means for embedding said data within a comment section of an HTML file;

means for sending said HTML file to said receiving side host;

means for extracting said embedded data from said comment section of said HTML file;

means for preparing a local copy of said extracted data on said receiving side host.

11. An apparatus as defined in claim 10, further comprising:

means for launching an applet on said receiving side host that uses certain portions of said extracted data.

12. An apparatus as defined in claim 11, wherein said means for preparing said local copy includes means for locating said extracted data at a file path on said receiving side host, said means for launching said applet including means for passing a parameter including said file path of said local copy to said applet.

13. An apparatus as defined in claim 12, wherein said means for launching said applet includes:

means for creating a second HTML file with an instruction to launch said applet;

means for inserting said parameter in said second HTML file; and
means for invoking an application to load said HTML file.

14. An apparatus as defined in claim 10, wherein said means for preparing said data includes:
means for encrypting said data if required by a user; and
means for encoding said data in a predetermined format.

15. An apparatus as defined in claim 14, wherein said means for preparing said local copy of said extracted data includes:
means for decoding said data in accordance with said predetermined format; and
means for decrypting said data if said decoded data is encrypted.

16. An apparatus as defined in claim 10, further comprising:
means for determining whether a process for performing said extracting step exists on said receiving side host; and
means for downloading said process from an external source if said process does not exist on said receiving side host.

17. An apparatus as defined in claim 11, wherein said applet is contained within certain other portions of said extracted data.

18. A method of distributing data between a sending side host and a receiving side host, comprising:
receiving an HTML file sent by said sending side host in said receiving side host;
extracting embedded data from a comment section of said HTML file; and
preparing a local copy of said extracted data on said receiving side host.

19. A method as defined in claim 18, further comprising:
launching an applet on said receiving side host that uses certain portions of said extracted data.

20. A method as defined in claim 18, further comprising:

determining whether a process for performing said extracting step exists on said receiving side host; and

downloading said process from an external source if said process does not exist on said receiving side host.

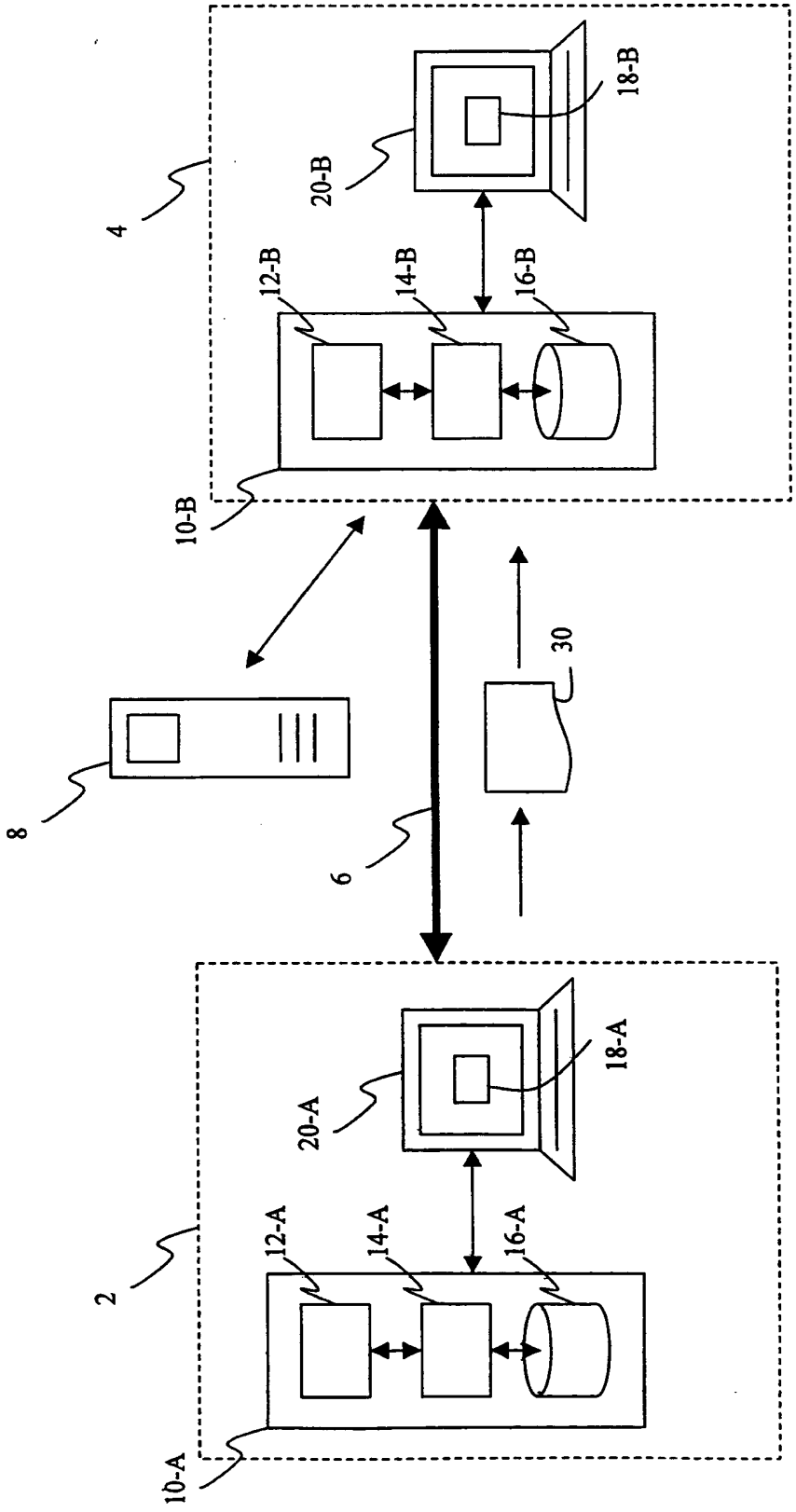


FIGURE 1

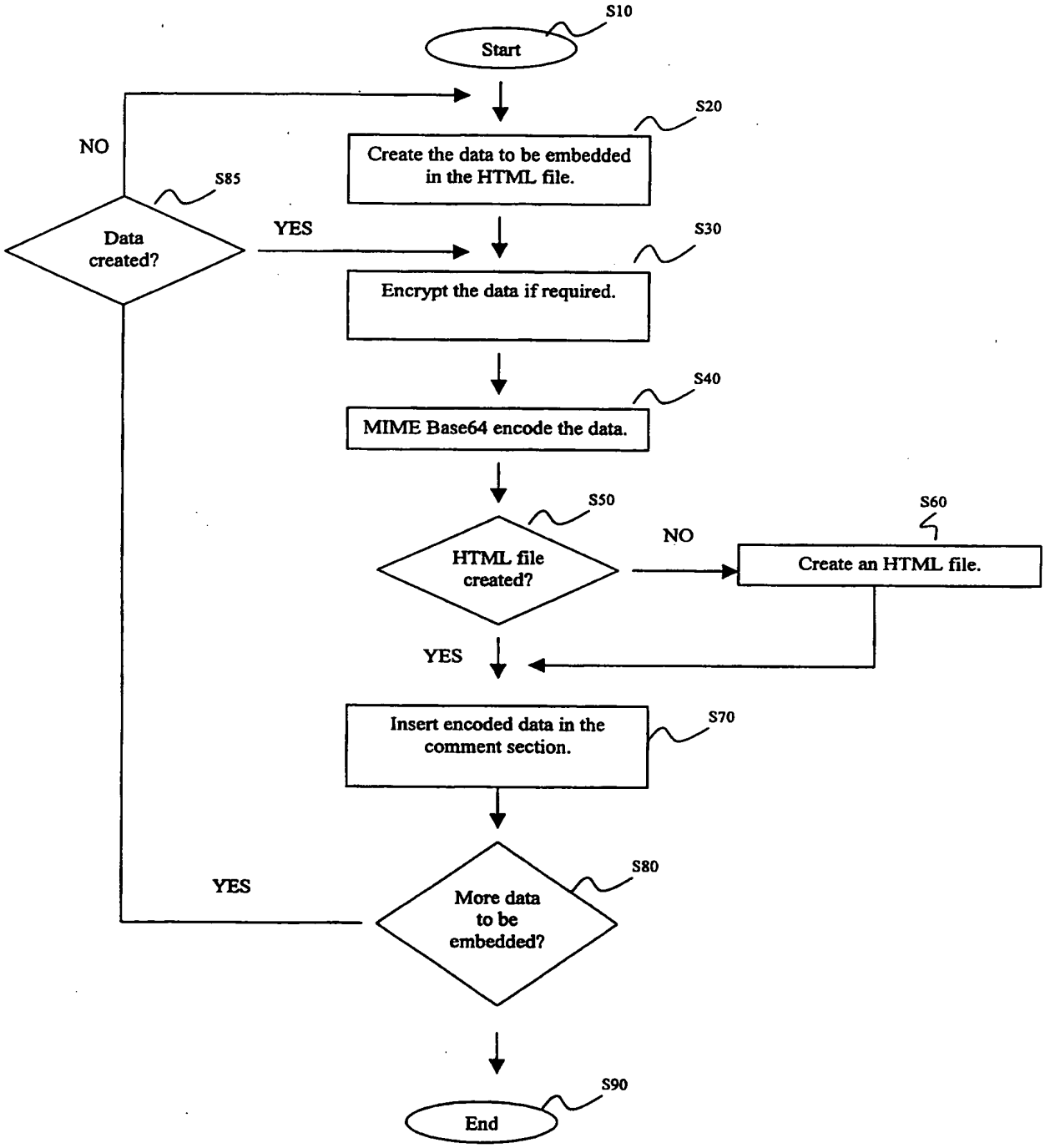

S10
Start
S20
Create the data to be embedded in the HTML file.
S85
Data created?
NO
YES
S30
Encrypt the data if required.
S40
MIME Base64 encode the data.
S50
HTML file created?
NO
S60
Create an HTML file.
YES
Insert encoded data in the comment section.
S70
S80
More data to be embedded?
YES
S90
End

FIGURE 2

```
<HTML>

50  if (Plug-in exists locally)
    {
        if (Plug-in version is incorrect)
            Download the correct plug-in version
    } else {
        Download the plug-in
    }

52  Execute the plug-in
    .
    .
54  <!Start Embedded Data 1
    .
    .       Data 1
    .
    End Embedded Data 1>

    <!Start Embedded Data 2
    .
    .       Data 2
    .
    End Embedded Data 2>
    .
    .
    .
    .
    <!Start Embedded Data N
    .
    .       Data N
    .
    End Embedded Data N>

</HTML>
```

FIGURE 3

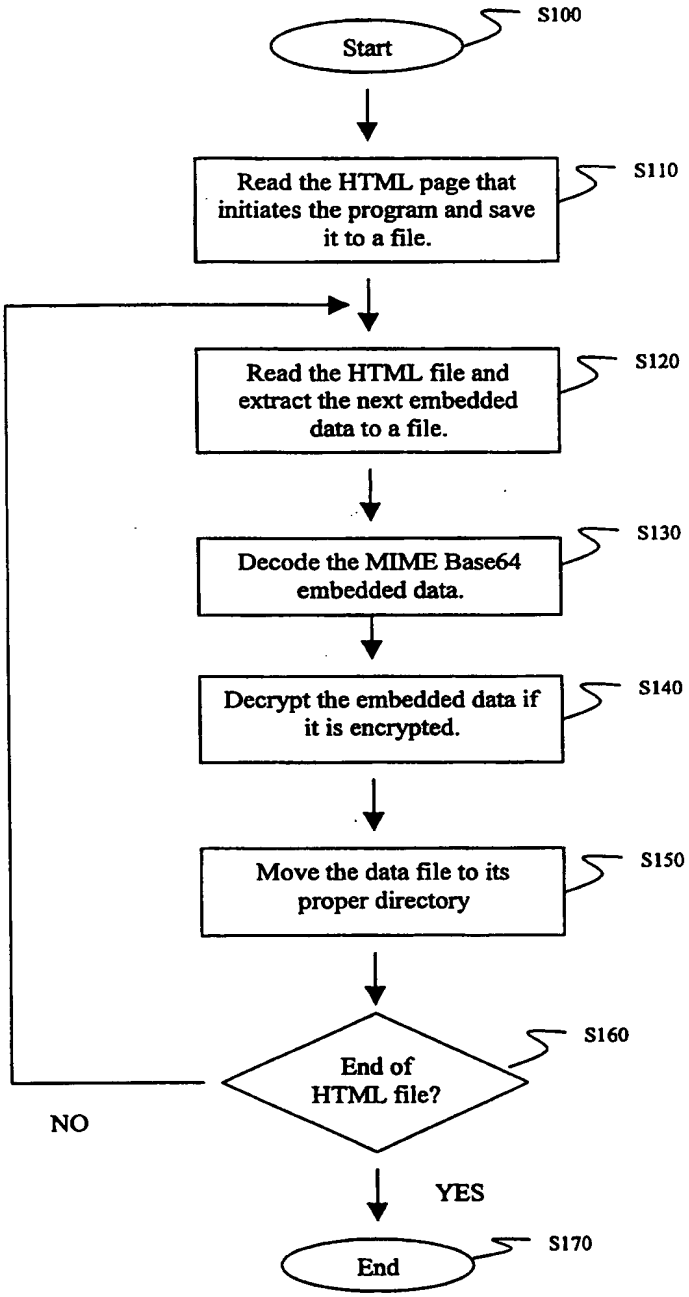

S100
Start
Read the HTML page that initiates the program and save it to a file.
S110
Read the HTML file and extract the next embedded data to a file.
S120
Decode the MIME Base64 embedded data.
S130
Decrypt the embedded data if it is encrypted.
S140
Move the data file to its proper directory
S150
End of HTML file?
S160
NO
YES
End
S170

FIGURE 4

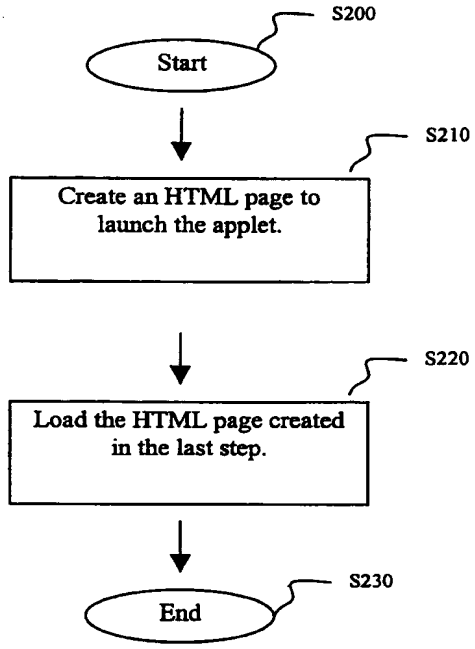

S200
Start
S210
Create an HTML page to launch the applet.
S220
Load the HTML page created in the last step.
S230
End

FIGURE 5

```
<HTML>

60 <APPLET archive="xxx.jar" code="yyy.class" width=w
   height=h>

62 <PARAM name="param1" value=val1>
   <PARAM name="param2" value=val2>
   ...
   ...
   <PARAM name="paramn" value=valn>
   </APPLET>
   </HTML>
```

FIGURE 6

```
<HTML>

Opening Envelope.  One moment please...

<SCRIPT>

function startDownload(minVersion) {

   var myMimetype = navigator.mimeTypes["application/x-envelope-plugin"];
   var trigger = netscape.softupdate.Trigger;

   // If some version is already installed on this machine...
   if ( myMimetype )    {
   // Check existing plug-in version and download new version if
       necessary
   }

   // No version of Envelope Plug-in is currently installed on this
   // machine, so start the download
   else
      return trigger.StartSoftwareUpdate(
       "http://popx.postx.com/~vincent/EnvelopePlugin.jar",
       trigger.DEFAULT_MODE);

      return false;
  }
startDownload (0); // Install plugin

</SCRIPT>

<! Start the plug-in>
<EMBED type=application/x-envelope-plugin>

<!MIME-Base64 encoding>
<!Envelope Start
AYACAQAAAAAAgAAUG9zdFggUGVyc29uYWwgRW52ZWxvcGUAAAAAAAAAAAAAAAWh0dHA6Ly93
d3cucG9zdHguY29tAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
GQEAAQAAAAAAC3J1ZmluYW4uZ2lmAQAAAABYAAAAFwEAAQABAAAACW9mZmVyLmdpZgEAAAAA
WAAAABYBAAEAAgAAAAhsb2FuLmdpZgEAAAAAYwAAAIYCAGwA5gBaAEQBAAAAeFgAAAAYAQAB
AAAAAAAKZmxpZ2h0LmdpZgEAAAAAWAAAABcBAAEAAQAAAAltaWxlcy5naWYBAAAAAFgAAAAe
AgcBAAIAAAAQRG1hbG9nU3VydmV5LmphcgEAAAAAWAAAABcCBgEAAAAAAAlwYXJhbS50eHQB
AAAAAAA=
Envelope End>

<!Envelope Vixel Start
AYACAQAAAAAAgAAUG9zdFggUGVyc29uYWwgRW52ZWxvcGUAAAAAAAAAAAAAAAWh0dHA6Ly93
d3cucG9zdHguY29tAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AgcBAAIAAAAQRG1hbG9nU3VydmV5LmphcgEAAAAAWAAAABcCBgEAAAAAAAlwYXJhbS50eHQB
AAAAAAA=
Envelope Vixel End>

</HTML>
```


FIGURE 7

```
<HTML>

64 <APPLET archive="EnvelopeApplet.jar"
   code="EnvelopeApplet.class" width=1 height=1>

   <PARAM name="envelopeName"
   value="D:\NETSCAPE\COMMUNICATOR\PROGRAM\Plugins\PXEvlp
   \Temp\tmpD1B1.evp">

   <PARAM name="tempDirectory"
   value="D:\NETSCAPE\COMMUNICATOR\PROGRAM\Plugins\PXEvlp
   \Temp\">

66 <PARAM name="systemDirectory" value="C:\WINDOWS\">

   <PARAM name="jarDirectory"
   value="D:\NETSCAPE\COMMUNICATOR\PROGRAM\Plugins\PXEvlp
   \">

   <PARAM name="selfExePath" value="">

   <PARAM name="javaInterpreter" value="">

   </APPLET>

   </HTML>
```

FIGURE 8

INTERNATIONAL SEARCH REPORT

Inter onal Application No
PCT/US 98/27420

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F9/46 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	NETSCAPE COMMUNICATIONS CORPORATION: "Javascript 4.0 Guide - Netscape Navigator " 1996 , NETSCAPE, MOUNTAIN VIEW, CALIFORNIA , USA XP002103707 see page 27, line 9 - page 31, line 4 ---	1,2, 8-11,18, 19
A	LAWRENCE, SCOTT: "Efficient Implementation of WWW Service in Embedded Systems" AGRANAT SYSTEMS, INC.,1997, pages 1-7, XP002103689 http://www.agranat.com/emweb_wp1.htm see page 2, line 20 - page 3, line 39 --- -/--	1,2, 8-11,18, 19

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 May 1999	10/06/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Bijn, K

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 98/27420

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	HAMMER J ET AL: "Extracting semistructured information from the Web" PROCEEDINGS OF THE WORKSHOP ON MANAGEMENT OF SEMI-STRUCTURED DATA, PROCEEDINGS OF WORKSHOP ON MANAGEMENT OF SEMI-STRUCTURED DATA, TUCSON, AZ, USA, 16 MAY 1997, pages 18-25, XP002103690 1997, Murray Hill, NJ, USA, AT & T Labs - Research, USA see abstract see page 2, line 16 - last line see page 5, paragraph 2.2 - page 6 ---	1,9,10, 18
A	EP 0 718 761 A (SUN MICROSYSTEMS INC) 26 June 1996 see abstract see claims 1-4 -----	7

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

information on patent family members

Inte ional Application No
PCT/US 98/27420

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0718761 A	26-06-1996	US 5630066 A	13-05-1997
		JP 8263447 A	11-10-1996
		US 5815661 A	29-09-1998